(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 437 658 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
14.07.2004 Bulletin 2004/29

(51) Int Cl.[7]: **G06F 11/14**, G06F 11/20

(21) Application number: **04003916.6**

(22) Date of filing: **28.07.1999**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE

(30) Priority: **25.08.1998 US 139257**

(62) Document number(s) of the earlier application(s) in accordance with Art. 76 EPC:
99937596.7 / 1 153 346

(71) Applicant: **Network Appliance, Inc.**
Sunnyvale, California 94089 (US)

(72) Inventors:
- **Schoenthal, Scott**
  San Ramon, CA 94583 (US)
- **Rowe, Alan**
  San Jose, CA 95123 (US)
- **Kleiman, Steven**
  Los Altos, CA 94022 (US)

(74) Representative: **Leeming, John Gerard**
J.A. Kemp & Co.,
14 South Square,
Gray's Inn
London WC1R 5JJ (GB)

Remarks:
This application was filed on 20 - 02 - 2004 as a divisional application to the application mentioned under INID code 62.

(54) Coordinating persistent status information with multiple file servers

(57) The invention provides a storage system, and a method for operating a storage system, that provides for relatively rapid and reliable takeover among a plurality of independent file servers. Each file server maintains a reliable communication path to the others. Each file server maintains its own state in reliable memory. Each file server regularly confirms the state of the other file servers. Each file server labels messages on the redundant communication paths, so as to allow other file servers to combine the redundant communication paths into a single ordered stream of messages. Each file server maintains its own state in its persistent memory and compares that state with the ordered stream of messages, so as to determine whether other file servers have progressed beyond the file server's own last known state.

Each file server uses the shared resources (such as magnetic disks) themselves as part of the redundant communication paths, so as to prevent mutual attempts at takeover of resources when each file server believes the other to have failed. Each file server provides a status report to the others when recovering from an error, so as to prevent the possibility of multiple file servers each repeatedly failing and attempting to seize the resources of the others.

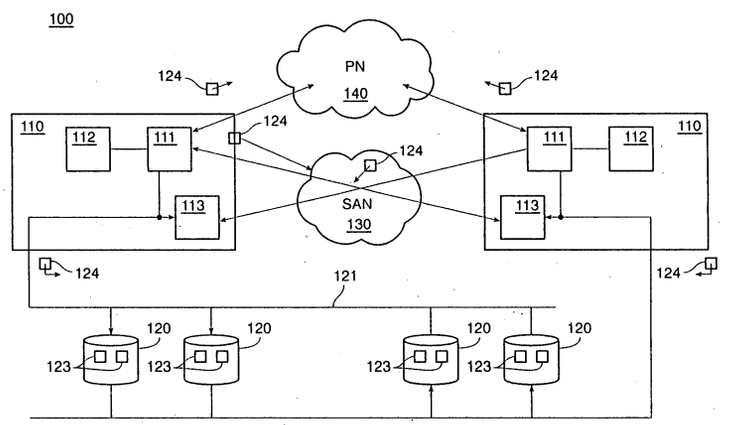

FIG. 1

EP 1 437 658 A2

Description

[0001] The invention relates to computer systems.

[0002] Computer storage systems are used to record and retrieve data. It is desirable for the services and data provided by the storage system to be available for service to the greatest degree possible. Accordingly, some computer storage systems provide a plurality of file servers, with the property that when a first file server fails, a second file server is available to provide the services and the data otherwise provided by the first. The second file server provides these services and data by takeover of resources otherwise managed by the first file server.

[0003] One problem in the known art is that when two file servers each provide backup for the other, it is important that each of the two file servers is able to reliably detect failure of the other and to smoothly handle any required takeover operations. It would be advantageous for this to occur without either of the two file servers interfering with proper operation of the other. This problem is particularly acute in systems when one or both file servers recover from a service interruption.

[0004] Accordingly, it would be advantageous to provide a storage system and a method for operating a storage system, that provides for relatively rapid and reliable takeover among a plurality of independent file servers. This advantage is achieved in an embodiment of the invention in which each file server (a) maintains redundant communication paths to the others, (b) maintains its own state in persistent memory at least some of which is accessible to the others, and (c) regularly confirms the state of the other file servers.

[0005] The invention provides a storage system and a method for operating a storage system, that provides for relatively rapid and reliable takeover among a plurality of independent file servers. Each file server maintains a reliable (such as redundant) communication path to the others, preventing any single point of failure in communication among file servers. Each file server maintains its own state in reliable (such as persistent) memory at least some of which is accessible to the others, providing a method for confirming that its own state information is up to date, and for reconstructing proper state information if not. Each file server regularly confirms the state of the other file servers, and attempts takeover operations only when the other file servers are clearly unable to provide their share of services.

[0006] In a preferred embodiment, each file server sequences messages on the redundant communication paths, so as to allow other file servers to combine the redundant communication paths into a single ordered stream of messages. Each file server maintains its own state in its persistent memory and compares that state with the ordered stream of messages, so as to determine whether other file servers have progressed beyond the file server's own last known state. Each file server uses the shared resources (such as magnetic disks) themselves as part of the redundant communication paths, so as to prevent mutual attempts at takeover of resources when each file server believes the other to have failed.

[0007] In a preferred embodiment, each file server provides a status report to the others when recovering from an error, so as to prevent the possibility of multiple file servers each repeatedly failing and attempting to seize the resources of the others.

Figure 1 shows a block diagram of a multiple file server system with coordinated persistent status information.

Figure 2 shows a state diagram of a method of operation for a multiple file server system with coordinated persistent status information.

[0008] In the following description, a preferred embodiment of the invention is described with regard to preferred process steps and data structures. However, those skilled in the art would recognize, after perusal of this application, that embodiments of the invention may be implemented using one or more general purpose processors (or special purpose processors adapted to the particular process steps and data structures) operating under program control, and that implementation of the preferred process steps and data structures described herein using such equipment would not require undue experimentation or further invention.

[0009] In a preferred embodiment, the file server system, and each file server therein, operates using inventions described in the following patent applications:

- International Application No. PCT/US99/05071, filed March 8, 1999, in the name of Network Appliance Inc., titled "Highly Available File Servers," International Publication No. WO 99/46680.

[0010] This application is referred to as the "Clustering Disclosure."

[0011] In a preferred embodiment, each file server in the file server system controls its associated mass storage devices so as to form a redundant array, such as a RAID storage system, using inventions described in the following patent applications:

- Application Serial No. 08/471,218, filed June 5, 1995, in the name of inventors David Hitz et al., titled "A Method for Providing Parity in a Raid Sub-System Using Non-Volatile Memory", now U.S. Patent No. 5,948,110, issued September 7,1999;
- Application Serial No. 08/454,921, filed May 31, 1995, in the name of inventors David Hitz et al., titled "Method for Maintaining Consistent States of a File System and for Creating User-Accessible Read-Only Copies of a File System," now U.S. Patent No. 5,819,292 issued October 6, 1998;

- Application Serial No. 08/464,591, filed May 31, 1995, in the name of inventors David Hitz et al., titled "Method for Allocating Files in a File System Integrated with a Raid Disk Sub-System", now U.S. Patent No. 6,038,570, issued March 14, 2000.

[0012] These applications are collectively referred to as the "WAFL Disclosures."

System Elements

[0013] Figure 1 shows a block diagram of a multiple file server system with coordinated persistent status information.

[0014] A system 100 includes a plurality of file servers 110, a plurality of mass storage devices 120, a SAN (system area network) 130, and a PN (public network) 140.

[0015] In a preferred embodiment, there are exactly two file servers 110. Each file server 110 is capable of acting independently with regard to the mass storage devices 120. Each file server 110 is disposed for receiving file server requests from client devices (not shown), for performing operations on the mass storage devices 120 in response thereto, and for transmitting responses to the file server requests to the client devices.

[0016] For example, in a preferred embodiment, the file servers 110 are each similar to file servers described in the Clustering Disclosure.

[0017] Each of the file servers 110 includes a processor 111, program and data memory 112, and a persistent memory 113 for maintaining state information across possible service interruptions. In a preferred embodiment, the persistent memory 113 includes a nonvolatile RAM.

[0018] The mass storage devices 120 preferably include a plurality of writeable magnetic disks, magneto-optical disks, or optical disks. In a preferred embodiment, the mass storage devices 120 are disposed in a RAID configuration or other system for maintaining information persistent across possible service interruptions.

[0019] Each of the mass storage devices 120 are coupled to each of the file servers 110 using a mass storage bus 121. In a preferred embodiment, each file server 110 has its own mass storage bus 121. The first file server 110 is coupled to the mass storage devices 120 so as to be a primary controller for a first subset of the mass storage devices 120 and a secondary controller for a second subset thereof. The second file server 110 is coupled to the mass storage devices 120 so as to be a primary controller for the second subset of the mass storage devices 120 and a secondary controller for the first subset thereof.

[0020] The mass storage bus 121 associated with each file server 110 is coupled to the processor 11 1 for that file server 110 so that file server 110 can control mass storage devices 120. In alternative embodiments, the file servers 110 may be coupled to the mass storage devices 120 using other techniques, such as fiber channel switches or switched fabrics.

[0021] The mass storage devices 120 are disposed to include a plurality of mailbox disks 122, each of which has at least one designated region 123 into which one file server 110 can write messages 124 for reading by the other file server 110. In a preferred embodiment, there is at least one designated region 123, on each mailbox disk 122 for reading and at least one designated region 123 for writing, by each file server 110.

[0022] The SAN 130 is coupled to the processor 111 and to the persistent memory 113 at each of the file servers 110. The SAN 130 is disposed to transmit messages 124 from the processor 111 at the first file server 110 to the persistent memory 113 at the second file server 110. Similarly, the SAN 130 is disposed to transmit messages 124 from the processor 111 at the second file server 110 to the persistent memory 113 at the first file server 110.

[0023] In a preferred embodiment, the SAN 130 comprises a ServerNet connection between the two file servers 110. In alternative embodiments, the persistent memory 113 may be disposed logically remote to the file servers 110 and accessible using the SAN 130.

[0024] The PN 140 is coupled to the processor 111 at each of the file servers 110. The PN 140 is disposed to transmit messages 124 from each file server 110 to the other file server 110.

[0025] In a preferred embodiment, the PN 140 can comprise a direct communication channel, a LAN (local area network), a WAN (wide area network), or some combination thereof.

[0026] Although the mass storage devices 120, the SAN 130, and the PN 140 are each disposed to transmit messages 124, the messages 124 transmitted using each of these pathways between the file servers 110 can have substantially differing formats, even though payload for those messages 124 is identical.

Method of Operation

[0027] Figure 2 shows a state diagram of a method of operation for a multiple file server system with coordinated persistent status information.

[0028] A state diagram 200 includes a plurality of states and a plurality of transitions there between. Each transition is from a first state to a second state and occurs upon detection of a selected event.

[0029] The state diagram 200 is followed by each of the file servers 110 independently. Thus, there is a state for "this" file server 110 and another (possibly same, possibly different) state for the "the other" file server 110. Each file server 110 independently determines what transition to follow from each state to its own next state. The state diagram 200 is described herein with regard to "this" file server 110.

[0030] In a NORMAL state 210, this file server 110 has control of its own assigned mass storage devices 120.

[0031] In a TAKEOVER state 220. this file server 110 has taken over control of the mass storage devices 120 normally assigned to the other file server 110.

[0032] In a STOPPED state 230, this file server 110 has control of none of the mass storage devices 120 and is not operational.

[0033] In a REBOOTING state 240, this file server 110 has control of none of the mass storage devices 120 and is recovering from a service interruption.

NORMAL State

[0034] In the NORMAL state 210, both file servers 110 are operating properly, and each controls its set of mass storage devices 120.

[0035] In this state, each file server 110 periodically sends state information in messages 124 using the redundant communication paths between the two file servers 110. Thus, each file server 110 periodically transmits messages 124 having state information by the following techniques:

- Each file server 110 transmits a message 124 by copying that message to the mailbox disks on its assigned mass storage devices 120.
  In a preferred embodiment, messages 124 are transmitted using the mailbox disks by writing the messages 124 to a first mailbox disk and then to a second mailbox disk.
- Each file server 110 transmits a message 124 by copying that message 124, using the SAN 130, to its persistent memory 113 (possibly both its own persistent memory 113 and that for the other file server 110).
  In a preferred embodiment, messages 124 are transmitted using the SAN 130 using a NUMA technique.
  and
- Each file server 110 transmits a message 124 by transmitting that message 124, using the PN 140, to the other file server 110.
  In a preferred embodiment, messages 124 are transmitted using the PN 140 using encapsulation in a communication protocol known to both file servers 110, such as UDP or IP.
  Each message 124 includes the following information for "this" file server 110 (that is, the file server 110 transmitting the message 124):
- a system ID for this file server 110;
- a state indicator for this file server 110; In a preferred embodiment, the state indicator can be one of the following:

  (NORMAL) operating normally,

  (TAKEOVER) this file server 110 has taken over control of the mass storage devices 120,

  (NO-TAKEOVER) this file server 110 does not want the receiving file server to take over control of its mass storage devices 120, and

  (DISABLE) takeover is disabled for both file servers 110.

- a generation number Gi, comprising a monotonically increasing number identified with a current instantiation of this file server 110;
  In a preferred embodiment, the instantiation of this file server 110 is incremented when this file server 110 is initiated on boot-up. If any file server 110 suffers a service interruption that involves reinitialization, the generation number Gi will be incremented, and the message 124 will indicate that it is subsequent to any message 124 send before the service interruption.
  and
- a sequence number Si, comprising a monotonically increasing number identified with the current message 124 transmitted by this file server 110.

[0036] Similarly, each message 124 includes the following information for "the other" file server 110 (that is, the file server 110 receiving the message 124):

- a generation number Gi, comprising a monotonically increasing number identified with a current instantiation of the other file server 110;
  and
- a sequence number Si, comprising a monotonically increasing number identified with the most recent message 124 received from the other file server 110.

[0037] Each message 124 also includes a version number of the status protocol with which the message 124 is transmitted.

[0038] Since the file server 110 receives the messages 124 using a plurality of pathways, it determines for each message 124 whether or not that message 124 is "new" (the file server 110 has not seen it before), or "old" (the file server 110 has seen it before). The file server 110 maintains a record of the generation number Gi and the sequence number Si of the most recent new message 124. The file server 110 determines that the particular message 124 is new if and only if:

- its generation number Gi is greater than the most recent new message 124;
  or
- its generation number Gi is equal to the most recent new message 124 and its sequence number Si is greater than most recent new message 124.

[0039] If either of the file servers 110 determines that the message 124 is not new, that file server 110 can ig-

nore that message 124.

[0040] In this state, each file server 110 periodically saves its own state information using the messages 124. Thus, each file server 110 records its state information both on its own mailbox disks and in its own persistent memory 113.

[0041] In this state, each file server 110 periodically watches for a state change in the other file server 110. The first file server 110 detects a state change in the second file server 110 in one of at least two ways:

- The first file server 110 notes that the second file server 110 has not updated its state information (using a message 124) for a timeout period.
  In a preferred embodiment, this timeout period is two-half seconds for communication using the mailbox disks and one-half second for communication using the SAN 130. However, there is no particular requirement for using these timeout values; in alternative embodiments, different timeout values or techniques other than timeout periods may be used. and
- The first file server 110 notes that the second file server 110 has updated its state information (using one or more messages 124) to indicate that the second file server 110 has changed its state.

[0042] In a preferred embodiment, the second file server 110 indicates when it is in one of the states described with regard to each message 124.

[0043] If the first file server 110 determines that the second file server 110 is also in the NORMAL state, the NORMAL-OPERATION transition 211 is taken to remain in the state 210.

[0044] The first file server 110 makes its determination responsive to messages 124 it receives from the second file server 110. If there are no such messages 124 for a time period responsive to the timeout period described above (such as two to five times the timeout period), the first file server 110 decides that the second file server 110 has suffered a service interruption.

[0045] If the first file server 110 determines that the second file server 110 has suffered a service interruption (that is, the second file server 110 is in the STOPPED state 230), the TAKEOVER-OPERATION transition 212 is taken to enter the TAKEOVER state 220.

[0046] The TAKEOVER-OPERATION transition 212 can be disabled by a message 124 state indicator such as DISABLE or NO-TAKEOVER

[0047] In a preferred embodiment, either file server 110 can disable the TAKEOVER-OPERATION transition 212 responsive to (a) an operator command, (b) a synchronization error between the persistent memories 113, or (c) any compatibility mismatch between the file servers 110.

[0048] To perform the TAKEOVER-OPERATION transition 212, this file server 110 performs the following actions at a step 213:

- This file server 110 sends the message 124 state indicator TAKEOVER to the other file server 110, using including the reliable communication path (including the mailbox disks 122, the SAN 130, and the PN 140).
- This file server 110 waits for the other file server 110 to have the opportunity to receive and act on the TAKEOVER-OPERATION transition 212 (that is, to suspend its own access to the mass storage devices 120.
- This file server 110 issues disk reservation commands to the mass storage devices 120 normally assigned to the other file server 110.
- This file server 110 takes any other appropriate action to assure that the other file server 110 is passive.

[0049] If the takeover operation is successful, the TAKEOVER-OPERATION transition 212 completes and this file server enters the TAKEOVER state 220. Otherwise (such as if takeover is disabled), this file server 110 returns to the NORMAL state 210.

TAKEOVER State

[0050] In the TAKEOVER state 220, this file server 110 is operating properly, but the other file server 110 is not. This file server 110 has taken over control of both its and the other's mass storage devices 120.

[0051] In this state, this file server 110 continues to write messages 124 to the persistent memory 113 and to the mailbox disks 122, so as to preserve its own state in the event of a service interruption.

[0052] In this state, this file server 110 continues to control all the mass storage devices 120, both its own and those normally assigned to the other file server 110, until this file server 110 determines that it should give back control of some mass storage devices 120.

[0053] In a preferred embodiment, the first file server 110 makes its determination responsive to operator control. An operator for this file server. 110 determines that the other file server 110 has recovered from its service interruption. The GIVEBACK-OPERATION transition 221 is taken to enter the NORMAL state 210.

[0054] In alternative embodiments, the first file server 110 may make its determination responsive to messages 124 it receives from the second file server 110. If the second file server 110 sends messages 124 indicating that it has recovered from a service interruption (that is, it is in the REBOOTING state 240), the first file server 110 may initiate the GIVEBACK-OPERATION transition 221.

[0055] To perform the GIVEBACK-OPERATION tran-

sition 221, this file server 110 performs the following actions at a step 222:

- This file server 110 releases its disk reservation commands to the mass storage devices 120 normally assigned to the other file server 110.
- This file server 110 sends the message 124 state indicator NORMAL to the other file server 110, including using the mailbox disks 122, the SAN 130, and the PN 140.
- This file server 110 disables the TAKEOVER-OPERATION transition 212 by the other file server 110 until the other file server 110 enters the NORMAL state 210. This file server 110 remains at the step 222 until the other file server 110 enters the NORMAL state 210.

[0056] When the giveback operation is successful, the GIVEBACK-OPERATION transition 221 completes and this file server enters the NORMAL state 210.

STOPPED State

[0057] In the STOPPED state 230, this file server 110 has control of none of the mass storage devices 120 and is not operational.

[0058] In this state, this file server 110 performs no operations, until this file server 110 determines that it reboot.

[0059] In a preferred embodiment, the first file server 110 makes its determination responsive to operator control. An operator for this file server 110 determines that it has recovered from its service interruption. The REBOOT-OPERATION transition 231 is taken to enter the REBOOTING state 240.

[0060] In alternative embodiments, the first file server 110 may make its determination responsive to a timer or other automatic attempt to reboot. When this file server 110 determines that it has recovered from its service interruption, it attempts to reboot, and the REBOOT-OPERATION transition 231 is taken to enter the REBOOTING state 240.

REBOOTING State

[0061] In the REBOOTING state 240, this file server 110 has control of none of the mass storage devices 120 and is recovering from a service interruption.

[0062] In this state, the file server 110 attempts to recover from a service interruption.

[0063] If this file server 110 is unable to recover from the service interruption, the REBOOT-FAILED transition 241 is taken and this file server 110 remains in the REBOOTING state 240.

[0064] If this file server 110 is able to recover from the service interruption, but the other file server 110 is in the TAKEOVER state 220, the REBOOT-FAILED transition 241 is taken and this file server 110 remains in the REBOOTING state 240. In this case, the other file server 110 controls the mass storage devices 120 normally assigned to this file server 110, and this file server 110 waits for the GIVEBACK-OPERATION transition 221 before re-attempting to recover from the service interruption.

[0065] If this file server 110 is able to recover from the service interruption, and determines it should enter the NORMAL state 210 (as described below), the REBOOT-NORMAL transition 242 is taken and this file server 110 enters the NORMAL state 210.

[0066] If this file server 110 is able to recover from the service interruption, and determines it should enter the TAKEOVER state 210 (as described below), the REBOOT- TAKEOVER transition 243 is taken and this file server 110 enters the TAKEOVER state 210.

[0067] In a preferred embodiment, this file server 110 performs the attempt to recover from the service interruption with the following steps.

[0068] At a step 251, this file server 110 initiates its recovery operation.

[0069] At a step 252, this file server 110 determines whether it is able to write to any of the mass storage devices 120 (that is, if the other file server 110 is in the TAKEOVER state 220). If so, this file server 110 displays a prompt to an operator so indicating and requesting the operator to command the other file server 110 to perform the GIVEBACK-OPERATION transition 221.

[0070] This file server 110 waits until the operator commands the other file server 110 to perform a giveback operation, waits until the GIVEBACK-OPERATION transition 221 is complete, and proceeds with the next step.

[0071] At a step 253, this file server 110 determines the state of the other file server 110. This file server 110 makes this determination in response to its own persistent memory 113 and the mailbox disks 122. This file server 110 notes the state it was in before entering the REBOOTING state 240 (that is, either the NORMAL state 210 or the TAKEOVER state 220).

[0072] If this file server 110 determines that the other file server 110 is in the NORMAL state 210, it proceeds with the step 254. If this file server 110 determines that it had previously taken over all the mass storage devices 120 (that is, that the other file server 110 is in the STOPPED state 230 or the REBOOTING state 240), it proceeds with the step 255.

[0073] At a step 254, this file server 110 attempts to seize its own mass storage devices 120 but not those normally assigned to the other file server 110. This file server 110 proceeds with the step 256.

[0074] At a step 255, this file server 110 attempts to seize both its own mass storage devices 120 and those normally assigned to the other file server 110. This file server 110 proceeds with the step 256.

[0075] At a step 256, this file server 110 determines whether its persistent memory 113 is current with regard to pending file server operations. If not, this file server

110 flushes its persistent memory 113 of pending file server operations.

[0076] At a step 257, this file server 110 determines if it is able to communicate with the other file server and if there is anything (such as an operator command) preventing takeover operations. This file server 110 makes its determination in response to the persistent memory 113 and the mailbox disks 122.

[0077] At a step 258, if this file server 110 was in the NORMAL state 210 before entering the REBOOTING state 240 (that is, this file server 110 performed the step 254 and seized only its own mass storage devices 120), it enters the NORMAL state 210.

[0078] At a step 258, if this file server 110 was in the TAKEOVER state 220 before entering the REBOOTING state 240 (that is, this file server 1 10 performed the step 255 and seized all the mass storage devices 120, it enters the TAKEOVER state 220.

Alternative Embodiments

[0079] Although preferred embodiments are disclosed herein, many variations are possible which remain within the concept, scope, and spirit of the invention, and these variations would become clear to those skilled in the art after perusal of this application.

Claims

1. A file server (110), including:

an interface to a set of mass storage devices (120) and to at least one network (130, 140); and

a processor and controller (111) disposed to access said mass storage devices, to communicate messages (124) with at least a second file server (110) that has access to said mass storage devices, and to process state information about said server and said second file server;

wherein said messages are used to communicate said state information to and from said second file server, and wherein said messages are sent over plural different communication paths (120, 130, 140) including at least part (123) of said mass storage devices and said network.

2. A file server as in claim 1, wherein said part of said mass storage devices that are included in said communication paths further comprises one or more mailboxes stored on said mass storage devices.

3. A file server as in claim 1 or 2, wherein said plural different communication paths include at least one other network.

4. A file server as in claim 1, 2 or 3, wherein one of the servers can take over control of the mass storage devices from the other server.

5. A file server as in claim 4, wherein take over occurs if messages from the other server timeout or if messages from the other server indicate that the other server has changed state.

6. A file server as in claim 5, wherein different timeouts are used for the different communication paths.

7. A file server as in claim 4, 5 or 6, wherein said messages are used to prevent both servers from concurrently attempting to take over control of the mass storage devices.

8. A method of controlling a file server (110), comprising the steps of:

accessing a set of mass storage devices (120);

communicating messages (124) with at least a second file server (110) that has access to said mass storage devices; and

processing (111) state information about said server and said second file server;

wherein said messages are used to communicate said state information to and from said second file server, and wherein said messages are sent over plural different communication paths (120, 130, 140) including at least part (123) of said mass storage devices and a network.

9. A method as in claim 8, wherein said part of said mass storage devices that are included in said communication paths further comprises one or more mailboxes stored on said mass storage devices.

10. A method as in claim 8 or 9, wherein said plural different communication paths include at least one other network.

11. A method as in claim 8, 9 or 10, wherein one of the servers can take over control of the mass storage devices from the other server.

12. A method as in claim 11, wherein take over occurs if messages from the other server timeout or if messages from the other server indicate that the other server has changed state.

13. A method as in claim 12, wherein different timeouts are used for the different communication paths.

14. A method as in claim 11, 12 or 13, wherein said messages are used to prevent both servers from concurrently attempting to take over control of the mass

storage devices.

15. A computer program product comprising program code means that, when executed on a computer system, instruct the computer system to effect the steps of any one of claims 8 to 14.

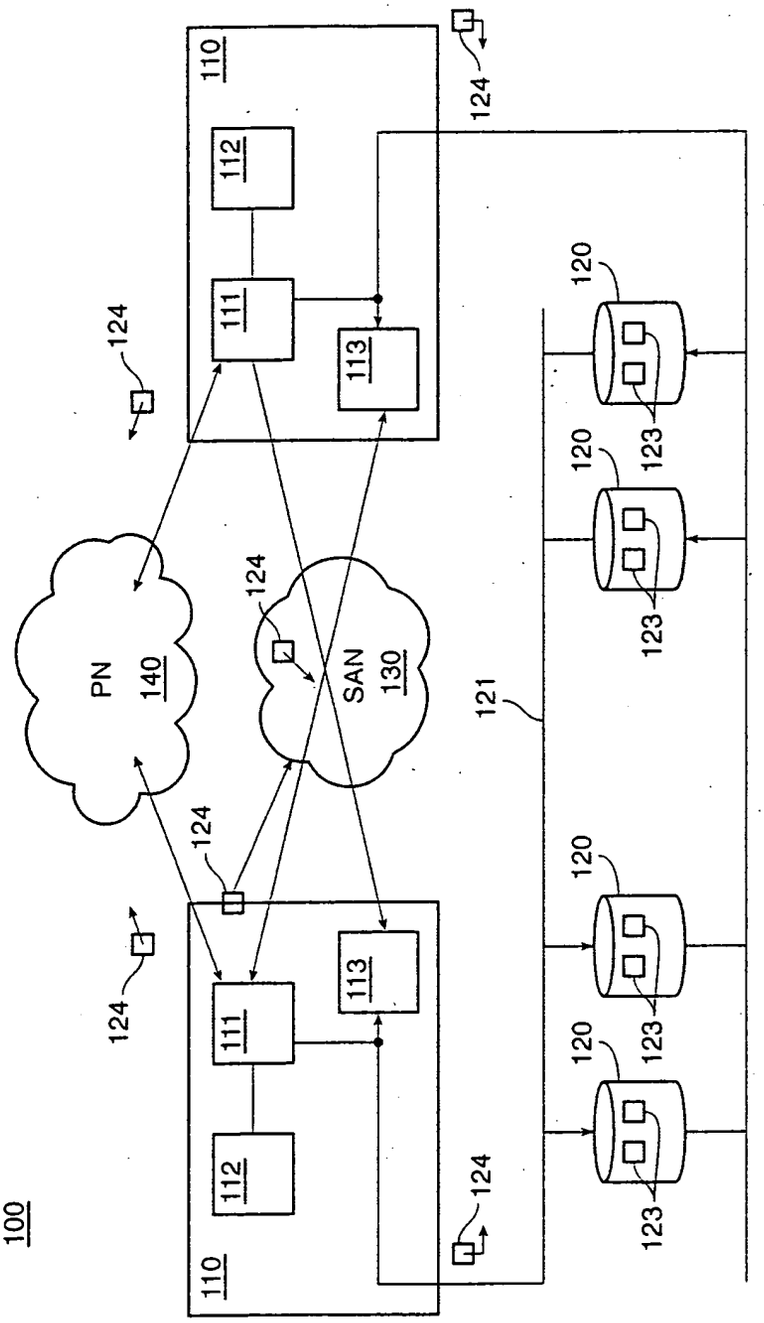

FIG. 1

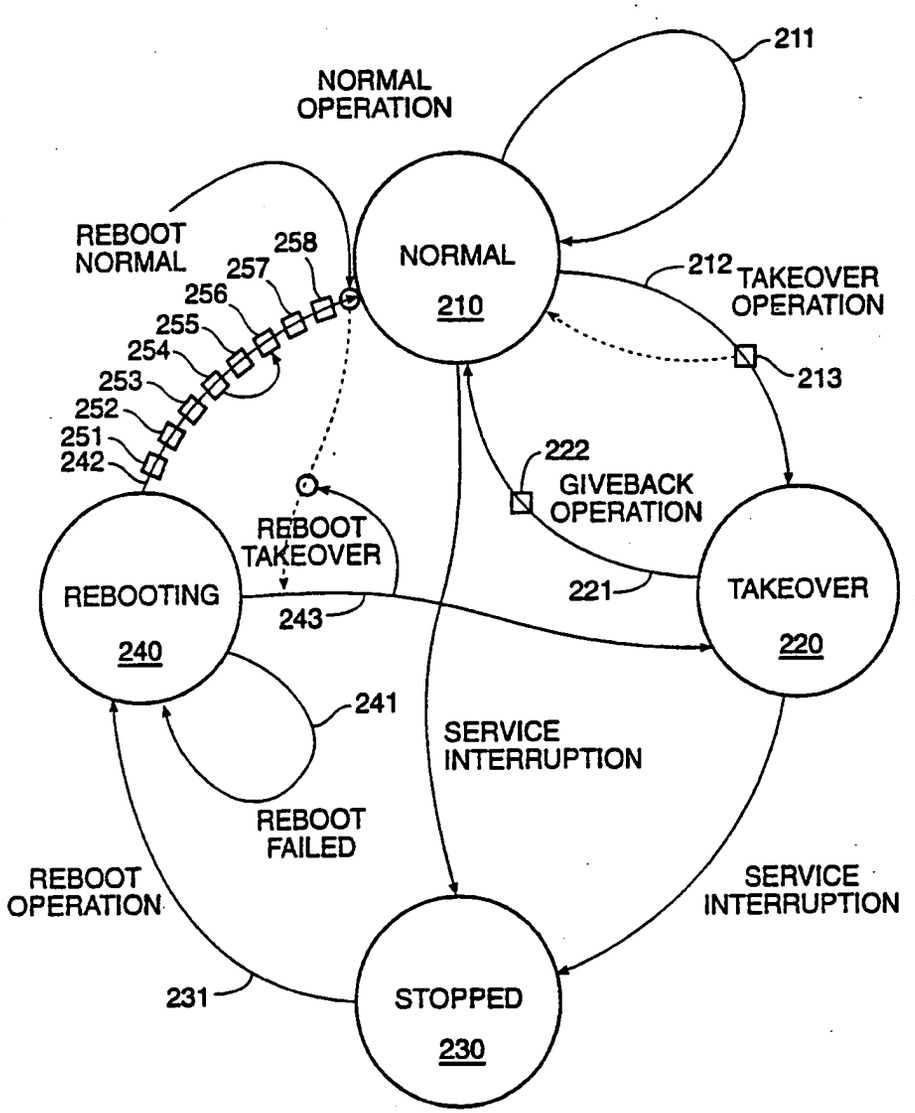
211
NORMAL OPERATION
NORMAL
210
REBOOT NORMAL
258
257
256
255
254
253
252
251
242
212
TAKEOVER OPERATION
213
222
GIVEBACK OPERATION
REBOOT TAKEOVER
REBOOTING
240
243
221
TAKEOVER
220
241
SERVICE INTERRUPTION
REBOOT FAILED
REBOOT OPERATION
SERVICE INTERRUPTION
231
STOPPED
230

FIG. 2